Underload Instabilities in Packet Networks With Flow Schedulers

Marco Ajmone Marsan, *Fellow, IEEE*, Mirko Franceschinis, *Student Member, IEEE*, Emilio Leonardi, *Member, IEEE*, Fabio Neri, *Member, IEEE*, and Alessandro Tarello, *Student Member, IEEE*

***Abstract*—Instability in packet-switching networks is normally associated with overload conditions, since queueing network models show that, in simple configurations, only overload generates instability. However, some results showing that instability can happen also in underloaded queueing networks began to appear about a decade ago. Underload instabilities can be produced by: 1) customer routes that visit the same queues several times; 2) variations of the customer service times at the different queues; and 3) complex scheduling algorithms. In this paper, we study, using fluid models and adversarial queueing theory, possible underload instabilities due to flow schedulers in packet networks, focusing on output queued switches with strict priority (SP) schedulers and Generalized Processor Sharing (GPS) schedulers. The considered scenarios always refer to acyclic packet routes and consider customer service times that vary only according to channel capacities, thus resembling the approaches being currently considered to provide QoS in the Internet. Our (in)stability results are rather surprising: SP schedulers appear to be more robust than GPS schedulers whenever exact information on the effective average packet flow rates is not available.**

***Index Terms*—Fluid models, QoS schedulers, queueing analysis, stability, switching.**

I. INTRODUCTION

SIMPLICITY has traditionally been one of the main design goals for packet networks, especially for the Internet. In particular, simplicity has for a long time been a characteristic of packet schedulers within IP routers since packets arriving at a router are normally inserted into simple FIFO queues. The desire for high-performance router architectures and QoS guarantees is now changing this picture, especially with regard to packet scheduling algorithms. Priority queues and/or Generalized Processor Sharing (GPS) queues [1], [2], in many packetized versions [3], are now widely accepted as useful tools for the treatment differentiation of packet flows referring to different end-user applications with different QoS requirements [4], [5].

When it comes to performance evaluation issues and considerations about possible instabilities due to the level of traffic at the router queues, the common belief is that only overload generates instability, while underloaded queues may induce delays longer than desired but will always remain stable. This general wisdom goes back to the models of packet-switching networks originally developed by Kleinrock [6] and based on Jackson queueing networks [7]. In Kleinrock's models, the packet[1] flows through the network were assumed to be Poisson, with rate λ_i packets per second over the ith channel of capacity C_i b/s, and the packet length was taken to be exponentially distributed with average μ^{-1} bit, so that the load at the ith queue is $\rho_i = (\lambda_i/\mu C_i)$, and the ith queue is stable as long as $\rho_i < 1$. Stability results for more general classes of queueing networks, such as BCMP networks [8] and Kelly networks [9], also confirmed the general result that only overload generates instability, and the same effect was due to the insensitivity of average queue sizes to the scheduling discipline for all work-conserving schedulers that treat customers independently of their service time in GI/GI/1 queues.

The first results showing that instability can happen also in underloaded queueing networks[2] started to appear about a decade ago [10]–[12] and identified some classes of queueing networks for which the backlog at some queue in the network can indefinitely grow also when such a queue is not overloaded. These underload instabilities are often produced either by customer routes that visit the same queues several times, by variations of the customer service times at the different queues, or by complex scheduling algorithms, which bear a significant resemblance to the flow schedulers today considered for packet-switching networks and to the scheduling algorithms used in IQ switches/routers [13]. In [14], it was shown that unstable underload behaviors can be observed for a class of open queueing networks called *re-entrant lines*, where customers visit the same queue several times, and all stations serve incoming customers according to a strict priority (SP) scheduling.

The first hints to the possible connections between underload instabilities in queueing networks and unstable behaviors in packet-switching networks appeared very recently in [15] and [16], where it was shown that underload instability phenomena can arise in networks of IQ switches with Maximum Weight Matching (MWM) schedulers [17], even in the case of acyclic packet routes, and service times that vary only according to channel capacities. To overcome these instabilities, Longest In Network (LIN), a new scheduling policy for IQ switches, was proposed in [15]. In [16], instead, a generalization of the MWM

Manuscript received February 14, 2003; revised October 23, 2003; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor B. Prabhakar. This work was supported by the Italian Ministry for University and Research through the TANGO project. A preliminary version of this paper was presented at the IEEE INFOCOM 2003, San Francisco, CA.

The authors are with the Dipartimento di Elettronica, Politecnico di Torino, 10129, Torino, Italy (e-mail: ajmone@polito.it; franceschinis@mail.tlc.polito.it; leonardi@mail.tlc.polito.it; neri@mail.tlc.polito.it; tarello@mail.tlc.polito.it).

Digital Object Identifier 10.1109/TNET.2004.838593

[1]Given the application domain, in this paper we use the terms "customer" and "packet" as synonyms.

[2]We say that a queueing network is underloaded when the average traffic loading each queue is less than the service capacity at the same queue. The average traffic at a queue is the product of the average customer arrival rate times the average customer service time.

scheduler was proposed; both of these scheduling policies guarantee stability under every underload traffic condition.

While [15] and [16] considered the possible underload instabilities due to scheduling in IQ switches, in this paper we discuss possible underload instabilities due to flow schedulers in networks of output queued packet switches, focusing both on SP schedulers and on GPS schedulers. The considered scenarios always refer to acyclic packet routes and consider customer service times that vary only according to channel capacities. With these assumptions, the considered scenarios bear a significant resemblance to the approaches being currently considered for QoS provisioning in the Internet, especially DiffServ [4], [5].

In particular, when considering GPS schedulers, we examine both the case of exact matching of the actual flow rates to the GPS rates and the more realistic case of inaccurate estimation of the actual flow rates. While the stability of any queueing network with GPS schedulers was proved for the case in which the actual flow rates do not exceed the GPS rates [2], [14], in this paper, we prove that a queueing network with GPS schedulers may be unstable when some of the actual flow rates exceed the GPS rates, thus confirming the conjecture formulated in [2].[3] Instead, when considering SP schedulers, we prove that all queueing networks are stable, provided that the priority ordering of packet flows does not change from a router (queue) to another. On the contrary, when the priority ordering of packet flows can change from a router to another, instability may arise, as proved in [11]. Finally, we prove that the combination of priority and FIFO schedulers (for example, due to packet flows that have identical priority at some routers) may lead to a weak form of instability for loads larger than 0.8. The fact that the combination of priority and FIFO may lead to instability is not surprising, due to the known possible instability of FIFO for loads above 0.85 [18], while the fact that the instability region expands to lower load values is instead quite remarkable. This finding is confirmed by a very recent result [19], showing that FIFO can be unstable for arbitrarily small network loads, thus further extending our result.

This paper is organized as follows. Section II introduces our definitions and notation. Section III provides a quick overview of the analytical tools that are used in this paper. Section III-A, in particular, briefly overviews some of the main results about underload instabilities in queueing networks. Section IV contains our main new results in the form of theorems with their proofs. Finally, Section V concludes the paper.

II. Definitions and Notation

We consider an acyclic network of J discrete-time physical queues[4] represented by row vector Q, whose jth component, $q^{(j)}$, $1 \leq j \leq J$, is a descriptor associated with the jth queue. However, all results in this paper apply also to networks of continuous-time queues.

The network of queues handles F customer flows. Customers belonging to flow f, $1 \leq f \leq F$, arrive at the network from outside, receive service at a number of queues, and leave the network. In most applications, the routing of customers belonging to flow f is deterministic; however, more general acyclic routings are allowed by our model.

Customers belonging to the same flow, and stored at the same physical queue, form a *virtual queue*. The number of virtual queues in a network is denoted by N. Of course, $N \leq FJ$, since routing is acyclic, and, at most, all customer flows visit all stations. In addition, $N \geq \max(F, J)$, since each flow must visit at least one station, and any station must be visited by at least one flow (otherwise the station can be removed from the network). We denote with $v^{(k)}$, $1 \leq k \leq N$, the kth virtual queue. Let $L(k) = j$ be the station location function that associates $v^{(k)}$ with the physical queue j at which customers are enqueued. $L^{-1}(j)$ is the counter-image of j through function $L(k)$. In general, $L^{-1}(j)$ returns a set of virtual queues of cardinality $|L^{-1}(j)|$.

Let $X_n = (x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(N)})$ be the row vector whose kth component $x_n^{(k)}$, $1 \leq k \leq N$, represents the number of customers at $v^{(k)}$ in the system at time n. We assume that X_0 is the null vector. Servers are associated with physical queues. A server provides service to customers queued at the virtual queues located at its physical queue according to a service policy. We suppose that the service time required by customer i of $v^{(k)}$ is a random variable $S_i^{(k)}$ distributed according to a discrete-time general distribution with average $s^{(k)}$; service times distributions are assumed to be bounded. We consider only nonpreemptive atomic service policies, i.e., policies that serve customers in an atomic fashion, with no service interruption. However, all our results also hold for work-conserving preemptive service policies.

The evolution of the number of customers at $v^{(k)}$ is described by $x_{n+1}^{(k)} = x_n^{(k)} + e_n^{(k)} - d_n^{(k)}$, where $e_n^{(k)}$ represents the number of customers that entered $v^{(k)}$ in time interval $(n, n+1]$, and $d_n^{(k)}$ is the number of customers departed from $v^{(k)}$ in $(n, n+1]$. $E_n = (e_n^{(1)}, e_n^{(2)}, \ldots, e_n^{(N)})$ is the vector of entrances in virtual queues, and $D_n = (d_n^{(1)}, d_n^{(2)}, \ldots, d_n^{(N)})$ is the vector of departures from virtual queues. With this notation, the system evolution equation can be written as

$$X_{n+1} = X_n + E_n - D_n. \tag{1}$$

The entrance vector E_n is sum of two terms: vector $A_n = (a_n^{(1)}, a_n^{(2)}, \ldots, a_n^{(N)})$, representing the customers arrived at the system from outside the network of queues (we also call them exogenous arrivals) and vector $R_n = (r_n^{(1)}, r_n^{(2)}, \ldots, r_n^{(N)})$ of recirculating customers; $r_n^{(k)}$ is the number of customers departed from some virtual queue and entered into $v^{(k)}$ in $(n, n+1]$.

The $N \times N$ matrix $P_n = [p_n^{(k,l)}]$ is the *routing matrix*, whose element $p_n^{(k,l)}$ represents the fraction of customers departing from $v^{(k)}$ in $(n, n+1]$ that enter $v^{(l)}$. Since routing is acyclic, P_n can be put in (upper or lower) triangular form by reordering rows and columns. Since in most applications the routing of each flow is deterministic, P_n belongs to the class of deterministic doubly substochastic matrices, i.e., matrices whose elements take values in the set {0,1} and whose lines (both rows and columns) comprise elements whose sums do not exceed 1. Thus, throughout this paper, we suppose that $P_n = P$ is deter-

[3]For the case in which some actual flow rates exceed the GPS rates, only a particular subclass of networks of queues with GPS schedulers was proved in [2] to be always stable in underload conditions.

[4]We use the terms "physical queue" and "station" as synonyms.

ministic, even if our models and most of the results can be easily extended to more general contexts.[5] By noting that $R_n = D_n P$, the evolution of virtual queues can be rewritten as

$$X_{n+1} = X_n + A_n - D_n(I - P) \tag{2}$$

where I denotes the identity matrix.

Finally, let us introduce the following norm definition.

Definition 1: Given a vector $Z \in \mathbb{R}^N$, $Z = (z^{(k)}, 1 \leq k \leq N)$, and a location function $L(k) = j$, from $1 \leq k \leq N$ to $1 \leq j \leq J$, with $J \leq N$, norm $\|Z\|_L$ is defined as

$$\|Z\|_L = \max_{j=1,\ldots,J} \left\{ \sum_{k \in L^{-1}(j)} \left| z^{(k)} \right| \right\}. \tag{3}$$

A. Traffic and System Stability Definitions

Definition 2: The external arrival process A_n is stationary if $\mathbb{E}[A_n] = \Lambda = (\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(N)})$, i.e., if $\mathbb{E}[A_n]$ does not depend on time interval $[n, n+1)$.

Let $\Pi = \Lambda(I - P)^{-1}$ be a vector whose kth component $\pi^{(k)}$ represents the average effective arrival rate at $v^{(k)}$. Let $W_n = (w_n^{(1)}, w_n^{(2)}, \ldots, w_n^{(N)})$ be the workload required at each virtual queue by customers that entered the network of queues in time interval $[n, n+1)$, i.e., the global amount of work required at each virtual queue by customers entering the network in time interval $[n, n+1)$. The average workload required at each virtual queue by customers that entered the system of queues in time interval $[n, n+1)$ is given by: $\Omega = \mathbb{E}[W_n] = (\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(N)})$, with $\omega^{(k)} = \pi^{(k)} s^{(k)}$. Here $S = (s^{(1)}, s^{(2)}, \ldots, s^{(N)})$ is the vector of the average service times at virtual queues.

Definition 3: A stationary traffic pattern is admissible if $\|\Omega\|_L < 1$.

Definition 4: A stationary arrival process A_n at virtual queues satisfies the strong law of large numbers if $\lim_{n\to\infty}(1/n)\sum_{i=0}^{n-1} A_i = \Lambda$ with probability 1 (w.p. 1).

Definition 5: A system of queues is *rate-stable* if $\lim_{n\to\infty}(X_n/n) = \lim_{n\to\infty}(1/n)\sum_{i=0}^{n}(E_i - D_i) = 0$ w.p.1.

Definition 6: A system of queues *achieves 100% throughput* if it is rate-stable under any admissible traffic pattern satisfying the strong law of large numbers.

Definition 7: A system of queues is *stable* if, for every $\epsilon > 0$, there exists an $M > 0$, such that, for every n: $\mathbb{P}(\|X_n\|_L > M) < \epsilon$.

Definition 8: A traffic pattern is $[\rho, \phi]$ regulated if, for every window $\mathcal{W}$ comprising ϕ consecutive time units, the total workload provided by customers entering the network in $\mathcal{W}$ satisfies $\|\sum_{n\in\mathcal{W}} W_n\|_L \leq \rho\phi$.

Note that any $[\rho, \phi]$ regulated traffic pattern with $\rho < 1$ and finite ϕ falls in the class of (ρ_0, σ) leaky-bucket constrained traffic patterns,[6] with $\rho_0 = \rho$ and $\sigma = \rho\phi$. In addition, any (ρ_0, σ) leaky-bucket constrained traffic pattern, with $\rho_0 < 1$ and finite σ, falls within the class of $[\rho, \phi]$ regulated traffic patterns for $\phi \geq \phi_{\min}$. To see this, it is enough to choose ρ such that $\rho_0 < \rho < 1$ and then to consider any $\phi \geq \phi_{\min} = (\sigma/(\rho - \rho_0))$. Thus, the class of all $[\rho, \phi]$ regulated traffic patterns with $\rho < 1$ and finite ϕ, and the class of all (ρ_0, σ) leaky-bucket constrained traffic patterns with $\rho_0 < 1$ and finite σ are coincident.

Definition 9: A system of queues is said to be *universally stable* if, under any $[\rho, \phi]$ regulated traffic pattern, with $\rho < 1$ and any $\phi < \infty$, the number of customers in the system remains finite.

Note that universal stability implies that delays in the queueing network are bounded under any leaky-bucket constrained traffic with finite burstiness.

The above stability definitions have different strengths and different implications in the application domain of packet-switching networks. The weaker stability Definitions are 5 and 6; they cannot guarantee the finiteness of queue lengths and are thus of lesser interest. Definition 7 is the standard definition for stability in queueing systems. Definition 9 is instead the stability definition deriving from adversarial queueing theory and is the strongest of our stability conditions. Stability under Definition 9 implies all other forms of stability, as shown in [20]; instability under Definition 7 implies instability under Definition 9.

III. Analytical Techniques

A. Previous Work

Two existing analytical tools were mainly applied in stability studies: the stochastic Lyapunov function methodology [21] and the fluid limit theory [14]. Both tools, and the latter in particular, can be applied to general networks of queues, under some mild regularity conditions on the stochastic processes describing the system. Using these tools, general necessary and sufficient conditions for the stability of networks of queues were identified, in the case of networks with two stations that operate under an SP service discipline [14]. In addition, it was proved that a large class of underloaded networks of FIFO queues[7] [22], as well as networks of GPS queues [14], where GPS rates are exactly matched to packet flow rates, are stable (according to Definitions 5 and 7).

As an alternative to the two cited stochastic techniques, a new analytical framework, called Adversarial Queueing Theory (AQT) [18], [23], was developed. Applying AQT, several interesting results were obtained. Common service disciplines, such as FIFO, LIFO, and Nearest-To-Go (NTG) (where priority is given to packets nearest to their destination), were proved not to be universally stable [18], [23], while other service disciplines, such as Longest-In-System (LIS) (where priority is given to packets first injected in the system), Shortest-In-System (SIS), and Furthest-To-Go (FTG), were proved to be universally stable [18].

An important stability result for underloaded GPS networks was obtained in [2] by applying Network Calculus [24], [25]

[5]It is possible to apply the fluid model methodology to all systems of queues forming an *open network* [6]. A network is said to be open if $\Delta = I + \mathbb{E}[P_n] + \mathbb{E}[P_n]^2 + \mathbb{E}[P_n]^3 + \ldots = (I - \mathbb{E}[P_n])^{-1}$ exists and is finite, i.e., $I - \mathbb{E}[P_n]$ is invertible for all n (the symbol $\mathbb{E}[\cdot]$ denotes the expectation of the quantity into brackets, while I is the identity matrix). It is, however, necessary to further assume that the routing matrix P_n satisfies the strong law of large numbers: $\lim_{n\to\infty}(1/n)\sum_{i=0}^{n-1} P_i = P$ with probability 1.

[6]A (ρ_0, σ) leaky-bucket constrained traffic pattern is a traffic pattern such that, in any window of t consecutive time units, the amount of workload arriving at any network server does not exceed $\rho_0 t + \sigma$.

[7]The result applies only to a restricted class of FIFO queueing networks: the class of generalized Kelly-type networks. In more general FIFO networks, it was shown that some underload instability may arise. However, since FIFO queueing networks modeling communication networks always fall in the class of generalized Kelly-type networks, we only consider such networks in this paper.

concepts. In [2], any underloaded network of queues implementing GPS schedulers was proved to be stable when fed by leaky-bucket constrained traffic flows, whenever the GPS rates assignment to flows satisfies the Consistent Relative Session Treatment (CRST) constraint. The CRST constraint implies that, for any pair of flows f_i and f_j whose paths are not disjoint, the ratios between GPS weights and packet actual rates are in the same order relation at every shared node.

B. Adversarial Queueing Theory

AQT [18], [23] takes a deterministic approach to define stability criteria for queueing networks in the sense that packet generation at the network inputs, packet service times, and packet routing are supposed to be deterministic.

In AQT, a queueing network is represented by a directed graph, where edges represent queues and nodes represent routing points. Packets originate at a node, follow some path consisting of a set of consecutive edges and nodes, and then leave the network at another node. Different paths may share some edges. Time is assumed to proceed by discrete steps, named time units. When two or more packets following different paths need to traverse the same edge, only one can be chosen in a time unit, while the others are forced to wait in the node. The decision about which packet has to be served in case of contention defines the service discipline at the nodes. Only work-conserving service disciplines are considered, i.e., service disciplines according to which a packet must be served at a node whenever some packets are waiting for service.

The basic idea of AQT is the following. At each time unit, an adversary injects in the network a set of packets, each one characterized by a particular path, which is defined when the packet is generated, and cannot be changed. Of course, the network can be flooded by packets if no restriction is set on packet generation processes. Therefore, for any $0 \leq \rho < 1$, a (ρ, ϕ) *adversary* injects a $[\rho, \phi]$ regulated traffic pattern into the network. The aim of a (ρ, ϕ) adversary is to find a deterministic traffic pattern under which the length of some queue grows toward infinity, and the network becomes unstable.

More precisely, let G identify a network (a graph), S be a service discipline, and A an adversary: system (G, S) is said to be *universally stable* if, for every initial condition on G, and adversary A, a constant integer M exists, such that any queue length is upper bounded by M. In this case, the queueing network G is universally stable under service policy S. A queueing network G is said to be *universally stable* if it is universally stable under any service discipline S and, similarly, a scheduling policy S is said to be *universally stable* if it is stable for all networks G. It is worth noting that the queueing network stability criterion provided by AQT is very tight, because it derives from a worst-case analysis under a wide class of deterministic arrival patterns.

C. Fluid Models

The evolution of a queueing network is traditionally described by (2), where D_n is a function of both X_n and the scheduling policy. An alternative expression of the evolution of the queue lengths vector can be provided in terms of the cumulative processes. Let:

- $\mathcal{A}(n)$ be the cumulative number of arrivals in time interval $[0, n)$, i.e., $\mathcal{A}(n) = \sum_{i=0}^{n} A_i$;
- $\mathcal{D}(n)$ be the cumulative number of departures in time interval $[0, n)$, i.e., $\mathcal{D}(n) = \sum_{i=0}^{n} D_i$;
- $\mathcal{T}(n)$ be the cumulative amount of work provided to virtual queues in time interval $[0, n)$.

With these definitions, the equation describing the queue lengths vector evolution becomes

$$X_{n+1} = X_0 + \mathcal{A}(n) - \mathcal{D}(n)[I - P] \tag{4}$$

where $\mathcal{D}(n)$ is a stochastic function of $\mathcal{T}(n)$.

By linearly interpolating X_n, $\mathcal{A}(n)$, $\mathcal{D}(n)$, and $\mathcal{T}(n)$, we can define their extensions to continuous time: $\hat{X}(t)$, $\hat{\mathcal{A}}(t)$, $\hat{\mathcal{D}}(t)$, and $\hat{\mathcal{T}}(t)$. We focus our attention on $\lim_{m\to\infty}(1/m)\hat{X}(mt) = X(t)$ which is called the *fluid limit* of the queue length vector (this limit exists under weak regularity conditions). Note that $X(0) = \lim_{m\to\infty}(1/m)\hat{X}(0) = 0$ whenever a finite number of customers is in the system at time 0 (as we assume throughout the paper).

It was shown in [14] that the fluid limit of the queue length vector exists with probability 1 if the arrivals and the service times cumulative processes satisfy the strong law of large numbers, i.e., $\lim_{m\to\infty}(\mathcal{A}(mn)/m) = \Lambda n$, and $\lim_{m\to\infty}(\sum_{i=1}^{m} S_i^{(k)}/m) = s^{(k)}$. In addition, the fluid limit is a continuous function which is derivable almost everywhere with probability 1 (i.e., it is derivable in all points of $t \in \mathbb{R}^+$, except for at most a set of null Lebesgue measure).

In order to study the behavior of $X(t)$, the following theorem [14] is fundamental.

Theorem 1: The fluid limit $X(t)$ is a solution of the fluid model differential equations, obtained by averaging the stochastic equations of the system evolution

$$X(t) = X(0) + \Lambda t - D(t)(I - P) \tag{5}$$

$$D(t) = T(t)\Gamma \tag{6}$$

where Γ is a diagonal matrix whose nonnull elements $\gamma^{(k)}$ are equal to the inverse of the average service times at the different virtual queues, and $T(t) = (t^{(1)}(t), t^{(2)}(t), \ldots, t^{(N)}(t))$ is a nondecreasing function describing the time spent serving the different virtual queues in time interval $[0, t)$, which must satisfy additional constraints that identify the service discipline at queues.

For example, if the service discipline at queue j is work-conserving, then $T(t)$ satisfies the following equation:

$$\int_0^{\infty} \left[\sum_{k\in L^{-1}(j)} x^{(k)}(t)\right] d\left[t - \sum_{h\in L^{-1}(j)} t^{(h)}(t)\right] = 0 \tag{7}$$

i.e., the server is working whenever there is some unit of work to be performed. Indeed, inside the integral, we have the sum of virtual queue lengths at station j, and the integral is performed with respect to the difference between time and the work performed at station j; if queues are empty, the integral is zero; if queues are nonempty, the time increments must equal the performed work, so that again the integral becomes zero.

The following results relate the behavior of the solution of the fluid model (5) to the behavior of the network of queues.

Theorem 2: If $X(t) = 0\ \forall t > 0$ is the *only* solution of the fluid model equations, under the initial condition $X(0) = 0$, then the system of queues is rate-stable [14]. In this case, the fluid model is said to be *weakly stable*.

Theorem 3: Consider a fluid Markov model (i.e., a process with continuous state space that satisfies the Markov property). If t^* exists, such that for all fluid solutions, $X(t) = 0 \; \forall t > t^*$ under any initial condition with $\|X(0)\|_L = 1$ (a finite constant, in general), then the system of queues is Harris recurrent [14] (which is an extension of the concept of recurrence for Markov chains). In this case, the fluid model is said to be *stable*.

Theorem 4: If there exists $t > 0$ such that $X(t) > 0$ for *every* solution of the fluid model (5), under the initial condition $X(0) = 0$, then the system is unstable (i.e., some queue lengths grow to infinity) with probability 1. In this case, the fluid model is said to be *weakly unstable*.

Theorem 5: Consider a fluid Markov model. If an initial condition exists with $\|X(0)\|_L = 1$, such that, for every solution of the fluid model equations, $\limsup_{t\to\infty} \|X(t)\|_L = \infty$, then the queueing system is not Harris recurrent, i.e., it is not stable with probability 1. In this case, the fluid model is said to be *unstable*.

Note that, for fluid models, the definitions of weak instability and instability are not the converse definitions of weak stability and stability, respectively. Nothing can be said about the behavior of the system of queues when the fluid model is neither (weakly) stable nor (weakly) unstable. This is due to the fact that, in general, fluid limits are only a subset of the solutions of the fluid model differential equations. Thus, it is difficult to isolate fluid limits, discarding fluid solutions that do not correspond to fluid limits, and then infer information on the queue system behavior when fluid solutions exhibit different behaviors (i.e., some solutions are null and some are not null).

Fluid models and AQT are quite different methodologies, however, some relationships between the two exist, and have been investigated (see [20]).

IV. New Results

In this section, we present our stability results for networks of queues. Each of the results has been obtained making use of one of the previously described methodologies, and thus the results may be based on different assumptions. In particular, stationarity of arrival processes is always assumed in theorems where fluid models are used, while this is not true when the AQT approach is taken.

We start by considering in the first subsection networks of GPS queues,[8] and then we consider SP networks. By using the fluid model methodology, it was previously shown that a network of queues with GPS schedulers achieves 100% throughput under any admissible traffic pattern when GPS weights are equal to or greater than the corresponding effective average arrival rates [14]. This result was further strengthened by showing that networks of GPS schedulers are universally stable under the same condition on flow rates [2].[9] For these reasons, GPS schedulers are often considered the optimal architectural solution for communication networks supporting traffic streams with QoS requirements. However, since neither any form of Call Admission Control (CAC), nor any type of ingress traffic shaping is currently implemented in packet-switching networks adopting the Internet protocol suite, it is difficult to obtain a precise knowledge of the average effective flow rates at network routers (and at their schedulers). The calibration of GPS weights at routers (and at GPS schedulers within routers) often relies on local traffic measurements or rough estimates. As a consequence, a mismatch between GPS weights and effective average rates is not only possible, but likely. In the next subsection, we discuss an example of a network of queues implementing GPS schedulers in which instability arises (according to Definition 7) under admissible traffic when the nominal flow rates are not exactly matched to the effective average flow rates. These instabilities are shown to persist also when adaptive sources reduce arrival rates upon packet losses.

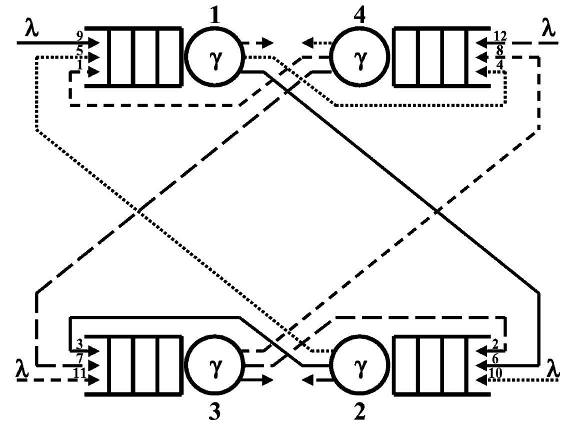

Fig. 1. Queueing network exhibiting instability.

We then focus our attention on networks of SP schedulers. We first prove that such networks are universally stable, provided that the priority ordering of packet flows does not change from a router (queue) to another (that is, if at a router flow 1 has higher priority than flow 2, the same happens also at all other routers). On the contrary, when the priority ordering of packet flows can change from a router to another, rate instability may arise, as proved in [10], [11], and [14]. Finally, we prove that networks of queues implementing a combination of SP and FIFO schedulers (for example, due to packet flows that have identical priority at some routers) are rate-stable, but not universally stable for loads larger than 0.8.

A. Networks of GPS Queues

Consider the queueing network shown in Fig. 1, which comprises four physical queues traversed by four different flows. Each flow enters the network at a physical queue, and follows a simple route[10] that traverses three physical queues. Three virtual queues are located at each physical queue, the first storing packets that have just entered the network, the second storing in-transit packets, and the last storing packets which are about to leave the network. The $N = 12$ virtual queues are numbered as shown in the figure. Each station adopts a GPS service discipline. In order to reduce the number of parameters of our model, we assume a perfect symmetry among the nominal flow rates: rate α is assigned at all stations to virtual queues storing packets

[8]GPS was originally proposed in [1], where a precise definition of this approach can be found.

[9]Later, in [26], the authors showed that GPS can be unstable even if the GPS rates equal the session rates; this does not contradict the result in [2] because the investigation in [26] is developed under more general conditions on input traffic: the corresponding model is defined as *temporary session model*, in contrast with the *permanent session model* [24], [25] under which [2] works.

[10]Routes are said to be "simple" when they are acyclic.

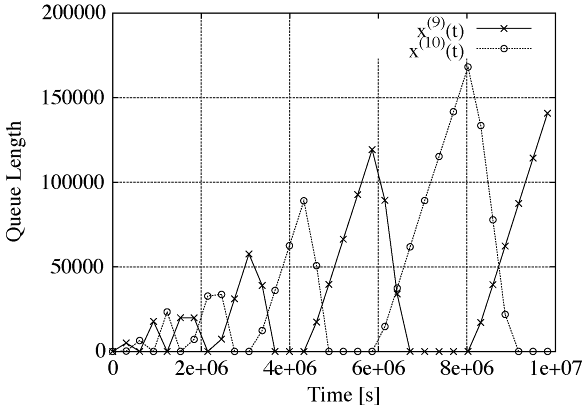

Fig. 2. Evolution of $x^{(9)}(t)$ and $x^{(10)}(t)$ by simulation with GPS weights not matched to flow rates.

that are about to leave the system, rate β is assigned to in-transit flows, and rate δ is assigned to flows entering the network.

We assume the server capacity at each station to be 1, so that relation $\alpha+\beta+\delta \leq 1$ must always hold. Fig. 2 reports the simulated evolution of the lengths of some virtual queues in the network of Fig. 1 as a function of time, assuming that all exogenous packet arrival processes (at $v^{(i)}$ with $i = 9, 10, 11, 12$) are Poisson with rate $\lambda^{(i)} = \lambda = (1/3.25)[s^{-1}]$, that service times are exponentially distributed random variables with average parameter $s^{(i)} = s = (1/\gamma) = 1[s]\ \forall i$, and that the GPS weights α, β, δ are, respectively, equal to 0.6, 0.3, and 0.1. The plots clearly show that the queueing network is unstable, in spite of the fact that no server in the network is overloaded (at each physical queue j, $\sum_{k\in L^{-1}(j)} \omega^{(k)} = (3/3.25) \approx 0.923$).

Theorem 6: Under admissible input flow rates, the fluid model of the queueing network in Fig. 1 can be unstable when the GPS weights α, β, and δ are such that $\delta < \alpha$ and $\delta < \beta$. In particular, instability arises when

$$\begin{cases} \frac{\lambda}{\gamma} \geq \frac{1-\alpha}{2(1-\alpha+\beta)}, & \text{if } \alpha > \beta \\ \frac{\lambda}{\gamma} \geq \frac{1-\beta}{2(1-\beta+\alpha)}, & \text{if } \alpha < \beta. \end{cases} \tag{8}$$

Proof: In order to write the system of deterministic differential equations that drive the fluid model associated with the queueing network, we need to introduce some notation. Given any function $f(t)$, let $\mathbf{U}_f$ be the Heaviside step function of $f(t)$, i.e., a function taking value 0 when $f(t) \leq 0$ and 1 when $f(t) > 0$. For each $v^{(k)}$, let k^* be the index of the virtual queue that comes before k in the packet route (if k is the first queue of the route, $k^* = 0$); let $c^{(k)}$ be the GPS weight associated with $v^{(k)}$. We can then write $\forall k = 1, \ldots, 12$, as two equations as follows:

$$\begin{cases} \dot{x}^{(k)}(t) = \gamma\left[\dot{t}^{(k^*)}(t) - \dot{t}^{(k)}(t)\right] \\ \dot{t}^{(k)}(t) = \mathbf{U}_{x^{(k)}} c^{(k)} r_{L(k)}(t) \\ \qquad + (1-\mathbf{U}_{x^{(k)}}) \min\left[\dot{t}^{(k^*)}(t), c^{(k)} r_{L(k)}(t)\right]. \end{cases} \tag{9}$$

The first says that the change in the length of the size of $v^{(k)}$ is obtained as the difference between the service rates at queues k^* and k; the second equation says that the service rate at $v^{(k)}$ is either (if the virtual queue is not empty) its GPS rate $c^{(k)}$ scaled by the largest possible coefficient $r_{L(k)}(t)$ accounting for empty virtual queues at the same station or (if $v^{(k)}$ is empty) the minimum between the service rate at $v^{(k^*)}$ and the product $c^{(k)} r_{L(k)}(t)$ Note that $\dot{t}^{(0)}(t)$ must be considered constant with respect to time t and equal to (λ/γ).

If the physical queue j is not empty, the following condition holds:

$$\sum_{h\in L^{-1}(j)} \dot{t}^{(h)}(t) = 1. \tag{10}$$

The second equation of the system (9) shows that $\dot{t}^{(h)}(t)$ is function of $r_j(t)$: this makes the computation of $\dot{t}^{(h)}(t)\forall h$ depend on the value of $r_j(t)$. The scaling factor $r_j(t)$ has to be maximized, always respecting condition (10); note the recursion: the values of $\dot{t}^{(h)}(t)$ depend on $r_j(t)$, whose computation is affected by $\sum_{h\in L^{-1}(j)} \dot{t}^{(h)}(t)$. These arguments can be formalized as

$$r_j(t) = \arg\max_{r_j(t)\geq 1} \left\{ \sum_{h\in L^{-1}(j)} \dot{t}^{(h)}(t) \leq 1 \right\}, \quad j = 1,2,3,4. \tag{11}$$

In spite of the apparent complexity of system (9), its solution is rather simple: indeed, the components $x^{(k)}(t)$ of the solution $X(t)$ are piecewise linear, while the components $\dot{t}^{(k)}(t)$ of the solution $\dot{T}(t)$ are piecewise constant; moreover, the discontinuity points correspond to time instants where some queue becomes empty.

We can for example solve (9) in the case $\alpha+\beta+\delta = 1$, $\gamma = 1\ [s^{-1}]$, $\alpha > \beta$, assuming that at time $t = 0$ the system is in the following initial condition:

$$x^{(k)}(0) = \begin{cases} q > 0, & k = 9, 12 \\ 0, & k \neq 9, 12. \end{cases}$$

The following values of $\dot{t}^{(k)}(t)$ can be computed:

$$\dot{t}^{(k)}(t) = \begin{cases} \frac{1-\alpha-\beta}{1-\alpha+\beta}, & k = 1,4,5,8,10,11 \\ \frac{\beta}{1-\alpha+\beta}, & k = 2,3,6,7 \\ \frac{\alpha+3\beta-1}{1-\alpha+\beta}, & k = 9,12. \end{cases} \tag{12}$$

A sketch of the computation follows. Due to symmetry, the values of $\dot{t}^{(k)}(t)$ at stations 1 and 4 must be the same, as well as at stations 2 and 3. We can thus write

$$\begin{aligned} \dot{t}^{(11)}(t) &= \dot{t}^{(10)}(t) = x \leq \lambda \\ \dot{t}^{(8)}(t) &= \dot{t}^{(5)}(t) = y \\ \dot{t}^{(6)}(t) &= \dot{t}^{(7)}(t) = z. \end{aligned}$$

Note that it must be $y \leq x$, because $v^{(5)}$ and $v^{(8)}$ are initially empty, and they receive x as input. We can also write that $\dot{t}^{(1)}(t) = \dot{t}^{(4)}(t) = y$, because $c^{(1)} = c^{(4)} = \alpha > (1/3) > \lambda$. Then we can deduce

$$\dot{t}^{(9)}(t) = \dot{t}^{(12)}(t) = 1 - 2y \tag{13}$$

because the whole service capacity is always used at a station if at least one virtual queue is not empty.

However, since $y \leq x \leq \lambda \leq (1/3)$, we get $1-2y \geq y$. This, together with $\beta > \delta$, forces $x = y$: indeed, according to (9), the solution $y < x$ is not admissible because it would cause $v^{(5)}$ and $v^{(8)}$ to receive less (or equal) service than $v^{(9)}$ and $v^{(12)}$ while growing, although their GPS weight β is strictly higher than δ, the GPS weight associated with $v^{(9)}$ and $v^{(12)}$.

Now, considering stations 2 and 3, we can write $\dot{t}^{(2)}(t) = \dot{t}^{(3)}(t) = z$. Indeed, it cannot be $\dot{t}^{(2)}(t) > z$, because $v^{(2)}$ receives z as input, and $x^{(2)}(0) = 0$; and it cannot be $\dot{t}^{(2)}(t) < z$, because $\alpha > \beta$.

To avoid that the service capacity at physical queues 2 and 3 exceeds 1, it must be $z < 1-2x$ (otherwise, the service capacity would be at least equal to $2 - 3x > 1$). As a result, $v^{(6)}$ and $v^{(7)}$ are forced to grow, because the fluid arrival rate is higher than the fluid departure rate. Thus, z and x can be determined as follows. First, we impose $2z + x = 1$, that is

$$z = \frac{1-x}{2}. \tag{14}$$

Then, recalling that $x \leq \lambda$ and comparing $v^{(6)}$ and $v^{(10)}$, we say that $x = \lambda$ is acceptable if $(x/\delta) < (z/\beta)$. Otherwise, we suppose $x < \lambda$ (this means that also $v^{(10)}$ is forced to grow) and obtain x by imposing $(x/\delta) = (z/\beta)$. We ignore the first alternative, which leads to all empty queues, and consider the case $x < \lambda$, which holds under condition

$$\lambda > \frac{1-\alpha-\beta}{1-\alpha+\beta}$$

which is obtained using (14).

The values reported in (12) can be easily obtained by substituting the value found for x in (13) and (14). Such values are valid until $x^{(9)}(t)$ and $x^{(12)}(t)$ become null. This happens at time

$$t_1 = q\left(\frac{\alpha+3\beta-1}{1-\alpha+\beta} - \lambda\right)^{-1}.$$

Considering the rates at which the fluid arrives and is drained at the different queues, the length of the queues at time t_1 is

$$x^{(k)}(t_1) = \begin{cases} q\frac{\lambda - \frac{1-\alpha-\beta}{1-\alpha+\beta}}{\frac{\alpha+3\beta-1}{1-\alpha+\beta} - \lambda}, & k = 10, 11 \\ q\frac{\alpha+2\beta-1}{(1-\alpha+\beta)\left(\frac{\alpha+3\beta-1}{1-\alpha+\beta} - \lambda\right)}, & k = 6, 7 \\ 0, & \text{otherwise.} \end{cases}$$

Proceeding as before, it is easy to verify that, in the new set of values for the $\dot{t}^{(k)}(t)$ after t_1, the only variation consists in $\dot{t}^{(9)} = \dot{t}^{(12)} = \lambda$.

The next relevant event is the emptying of $v^{(6)}$ and $v^{(7)}$ at time $t = t_1 + t_2$, where t_2 is given by the ratio of $x^{(6)}(t_1)$ [and $x^{(7)}(t_1)$] and the difference between the draining and arrival rates of fluid at those two queues. At time $t = t_1 + t_2$, the length of the queues is

$$x^{(k)}(t) = \begin{cases} q\frac{\lambda(1-\alpha+\beta)-(1-\alpha-\beta)}{\beta-\lambda(1-\alpha+\beta)}, & k = 10, 11 \\ 0, & k \neq 10, 11. \end{cases}$$

The details of the straightforward derivation are omitted for the sake of brevity.

By observing the evident symmetry between the network configuration that is reached at $t_1 + t_2$ and the one corresponding to the initial condition, it is possible to conclude that the quantities $x^{(10)}(t)$ and $x^{(11)}(t)$ (and the same is true for $x^{(9)}(t)$ and $x^{(12)}(t)$) grow to infinity, so that the fluid model is unstable, when

$$\frac{\lambda(1-\alpha+\beta)-(1-\alpha-\beta)}{\beta-\lambda(1-\alpha+\beta)} \geq 1.$$

This proves the assertion. With a similar procedure, it is possible to obtain the condition when $\beta > \alpha$. ∎

If the dynamics of the queueing network correspond to a Markov process, the instability of the fluid model implies the instability (according to Definition 7) of the queueing network. Thus, it is possible to conclude that the following theorem holds.

Theorem 7: A queueing network with GPS schedulers, whose evolution can be described by a Markov process, may be unstable (according to Definition 7) under admissible traffic when nominal flow rates at the stations are not matched to the effective average flow rates.

Note that Markov processes, i.e., processes satisfying the Markov property, possibly with continuous state space, are very general models, to which almost all queues and networks of queues of practical interest can be reduced. For example, the evolution of the unfinished work in a network of GI/GI/m queues can be described by a (complex) Markov process.

Of course, the instability of the fluid model automatically implies that the GPS service discipline is not universally stable when nominal flow rates are not matched to the effective average flow rates.

B. Networks of GPS Queues With Elastic Traffic

The instability of queueing networks with GPS schedulers we have just proved assumes that the exogenous packet arrival flows are stationary and satisfy the strong law of large numbers with admissible rates. In reality, packet networks in general and the Internet in particular are known to be subject to elastic input traffic flows (in the case of the Internet, the traffic elasticity is induced by the congestion control algorithms of TCP). It is thus interesting to discuss whether underload instabilities are mitigated by an adaptive behavior of traffic sources, as, for example, in the case of general additive-increase multiplicative-decrease (AIMD) source rate adaptation.

Consider again the network of Fig. 1, subject to idealized AIMD traffic sources, with AIMD coefficients $\Delta\theta_a = 0.01$ and $\Delta\theta_m = 0.5$, and with negligible source feedback delay (packet losses are immediately detected by sources): at every time step, the average of the exogenous flow arrival processes is increased by $\Delta\theta_a$ if the flow suffers no losses in the network, while the average is multiplied by $\Delta\theta_m < 1$ if losses occur. The GPS weights are $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 1 - \alpha - \beta = 0.1$. Further assume that the peak of each exogenous arrival rate is 0.33 of the server capacity, which is considered to be equal to 1, and that the buffering capacity at each virtual queue is finite

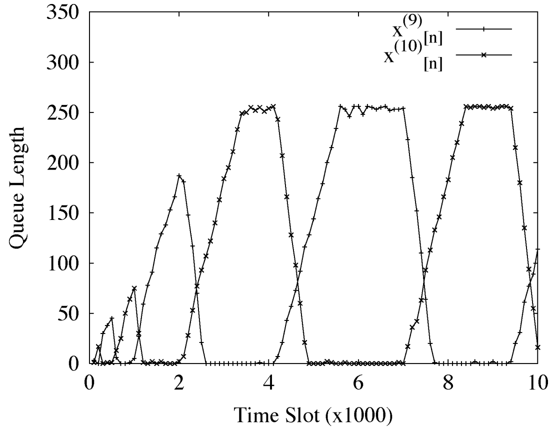

Fig. 3. Evolution of $x^{(9)}[n]$ and $x^{(10)}[n]$ by simulation with GPS service and adaptive sources.

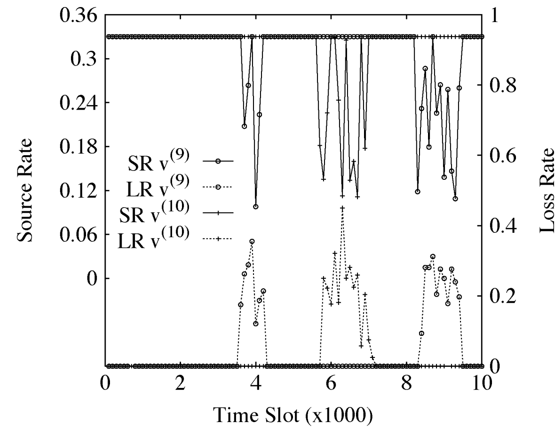

Fig. 4. Source and loss rates for the flows feeding $v^{(9)}$ and $v^{(10)}$ by simulation with GPS service and adaptive sources.

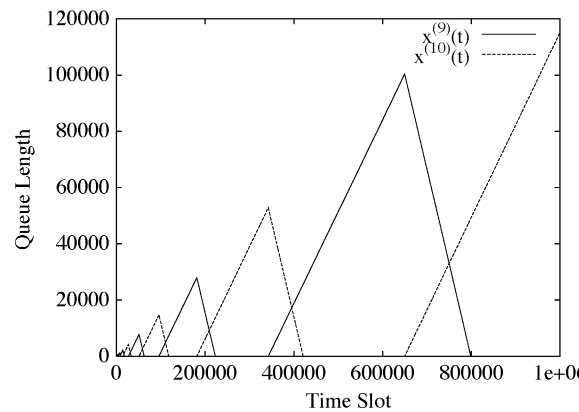

Fig. 5. Evolution of $x^{(9)}(t)$ and $x^{(10)}(t)$ by simulation with SP service.

and equal to 256. These parameter values are selected so that the peak rates of the four AIMD sources are not able to overload any physical queue (3×0.33 is less than 1).

In order to observe what happens in the network, we show in Fig. 3 the simulated evolution of $v^{(9)}$ and $v^{(10)}$, which closely resembles the behavior of the same queues fed by nonadaptive sources (see Fig. 2). The only major difference between the two plots is due to the finiteness of buffers, which does not permit an unlimited increase of the size of packets batches at queues: after an initial transient, the queues evolution tends to become periodic. In Fig. 4, we report for the same queues the evolution of the source and loss rates. The observed average throughput for each AIMD flow is 0.286. We can observe that the periodic behavior of queue lengths is generated by a sort of *pulsing behavior* of the AIMD sources; each source, indeed, interleaves periods in which it transmits at full speed, experiencing no losses, to periods in which its rate is drastically reduced, due to frequent losses experienced by the packet flows. It is worth noting that the packet loss process exhibits the same pulsing behavior of the source. Thus, by limiting the buffer capacity in the network and dynamically adapting the source rates to the network congestion conditions, we do not succeed in mitigating the underload instabilities that were observed with nonadaptive sources. However, the reader might wonder whether a different selection of source parameters would have permitted to improve the network performance. The answer is also negative in this case: by varying $\Delta\theta_a$ and $\Delta\theta_m$ and by introducing a feedback delay T, no throughput improvements were observed: in all cases, the network keeps exhibiting a similar pulsing behavior. The results of Figs. 3 and 4 were obtained with the fluid simulator, but similar behaviors were observed with a more accurate *ns* simulation model: in [27], we provided more insight on the behaviors due to AIMD source control.

C. Networks of Priority Queues

We consider in this section SP/FIFO queueing networks, defined as follows.

Definition 10: A *SP/FIFO queueing network* is a network of queues in which all virtual queues within the same physical queue are associated with a priority level; customers are served according to an SP nonpreemptive service discipline, i.e., customers are always extracted from one of the nonempty highest priority virtual queues. When the same priority is assigned to several virtual queues within the same station, ties are broken according to a FIFO service discipline.

We associate each $v^{(k)}$ with a priority index $p^{(k)}$ with $1 \leq p^{(k)} \leq |L^{-1}(L(k))|$, in such a way that lower indexes correspond to higher priorities. The same priority index is associated with virtual queues with the same priority residing at the same physical queue.

Dai [14] showed that instability phenomena can arise in SP queueing networks, thus reducing the maximum achievable throughput. However, instability was observed in queueing networks where either customers visit several times the same physical queue or the relative priority order assigned to flows is different in different physical queues. For example, in the queueing network depicted in Fig. 1 suppose SP is adopted by servers, and the priority index $p^{(k)}$ is, respectively, equal to: 1 if $v^{(k)}$ stores packets which are about to leave the network, 2 if it stores in-transit packets, and 3 if its packets have just entered the network. Simulation results (see Fig. 5) under the same admissible traffic pattern as in Section IV-A show that the network is unstable. However, in modern multiservice communication networks, packets normally follow simple routes, and priorities are usually assigned to packet flows in such a way that the relative priority order is the same at different nodes. For this reason, we specialize our investigation in this paper to a subclass of SP/FIFO queueing networks, called acyclic SP/FIFO networks.

A precise definition of acyclic SP/FIFO queueing networks, requires the concept of Priority Dependency Graph (PDG) induced by a priority assignment.

Definition 11: The PDG is the directed graph $G(V,E)$ satisfying the following properties:

- a vertex $v \in V$ corresponds to each network flow f;
- a directed edge $e \in E$ connects vertexes v_m and v_n (e: $v_m \to v_n$), corresponding respectively to flows f_m and f_n, if there exists a physical queue q traversed by flows f_m and f_n, such that $p^{(k_m)} < p^{(k_n)}$, where k_m and k_n are the virtual queues residing at q associated respectively with f_m and f_n.

Flow f_m is said to be the *father* of flow f_n if the PDG contains an edge from v_m to v_n; in this case, f_n is said to be the *child* of f_m.

Definition 12: An *acyclic SP/FIFO queueing network* is a SP/FIFO queueing network in which: 1) routes of flows are simple, i.e., all flows can visit at most once any physical queue and 2) PDG is acyclic.

We can now prove a very general and useful result.

Theorem 8: Each acyclic SP/FIFO queueing network is rate-stable under any admissible traffic pattern satisfying the strong law of large numbers.

Proof: The queueing network can be studied through (5) and (6) introduced in Section III-C. With the SP discipline, $T(t)$ satisfies, $\forall k$, the following variation of (7):

$$\int_0^\infty \left[\sum_{h\in L^{-1}_{k^+}(j)} x^{(h)}(t)\right] d\left[t - \sum_{h\in L^{-1}_{k^+}(j)} t^{(h)}(t)\right] = 0 \qquad (15)$$

where $L^{-1}_{k^+}(j)$ is the function that returns the set of virtual queues located at station j, whose priority index is not greater than $p^{(k)}$ (note that $v^{(k)}$ itself belongs to this set). According to (15), the server must work on the highest priority virtual queue, whenever there is some unit of work to be performed there.

Since the PDG is acyclic, we can associate an ordinal number o_f with $1 \le o_f \le F$ to each flow f in the network satisfying the following rule: for any pair of flows f_m and f_n, with f_m the father of f_n in the priority dependency graph, then $o_{f_m} < o_{f_n}$. Note that o_f induces an order relation among virtual queues. However, this order relation is quite weak, since all virtual queues corresponding to the same flow correspond to the same ordinal number. The previously defined order relation among virtual queues can be strengthened by associating an ordered pair of indexes with each virtual queue: the first index corresponds to the ordinal number of the flow that originates the considered virtual queue, while the second index, that we denote with $h^{(k)}$, represents the number of queues that the considered flow has traversed before $v^{(k)}$. Then, a global order relation $\mathcal{GO}$ can be introduced among virtual queues, according to the following rule: $v^{(k_m)}$ precedes $v^{(k_n)}$, iff $o_{f_m} < o_{f_n}$, or $o_{f_m} = o_{f_n}$ and $h^{(k_m)} < h^{(k_n)}$ along the path described by $f_m = f_n$.

Now we are ready to proceed with the proof. We first assume that there are no stations where two or more virtual queues have the same priority; we will relax this assumption later. Under this assumption, we claim the following.

Claim 1: Given any state vector $X(t)$ whose components are sorted in increasing order according to $\mathcal{GO}$, indicating with k_0 the index of the first nonnull component of $X(t)$ and $L(k_0) = j_0$, if t is a regular point for $X(t)$, then

$$\dot{x}^{(k_0)}(t) = \left(\sum_{k\in L^{-1}_{k_0^+}(j_0)} \omega^{(k)} - 1\right)\gamma^{(k_0)} = \chi < 0$$

where $\gamma^{(k_0)} = (1/s^{(k_0)})$ is the average service rate at $v^{(k_0)}$.

Proof: Let f_0 be the flow that originates k_0. For any empty $v^{(k)}$ in a regular point t, $\dot{x}^{(k)}(t) = 0$, as proved for a general fluid model in [14]; thus, the fluid departure rate equals the fluid arrival rate. By induction, it is immediate to verify that $\dot{e}^{(k)}(t) = \pi^{(k)}$ at each $v^{(k)}$ that follows only empty queues along the path described by its traffic flow. However, $v^{(k_0)}$, by construction, follows only empty queues along the path described by its traffic flow. Moreover, every virtual queue originated by a father of f_0 must be empty by construction. Thus, for all the higher priority $v^{(k)}$ colocated with k_0, $\dot{e}^{(k)}(t) = \dot{d}^{(k)}(t) = \pi^{(k)}$. We can write: $\dot{x}^{(k_0)}(t) = \dot{e}^{(k_0)}(t) - \dot{d}^{(k_0)}(t)$. But $\dot{e}^{(k_0)}(t) = \pi^{(k_0)}$, and $\dot{t}^{(k_0)}(t)$ is equal to 1 minus the sum of $\omega^{(m)}$ for all queues m residing at physical queue j_0, but having a higher priority than queue k_0. Hence, $\dot{d}^{(k_0)}(t)$ is equal to $\gamma^{(k_0)}$ minus the sum of $\pi^{(m)}$ at colocated, higher priority queues. The claim follows immediately, taking into account the fact that $\sum_{k\in L^{-1}_{k_0^+}(j_0)} \omega^{(k)} < 1$. $\square$

Consider now another nonempty $v^{(m)}$. The maximum amount of fluid which can arrive at it in a unit of time is bounded, due to the finiteness of the service capacity of all queues that come before $v^{(m)}$. Thus, there exists a constant C such that $|\dot{x}^{(m)}(t)| < C$, $\forall m$. In particular, we can take $C > |\chi|$. Consider the Lyapunov function $\mathcal{L}(X(t)) = \sum_{k=1}^{N} x^{(k)}(t)(|\chi|/2C)^k$. To prove stability, we must verify that, whenever $X(t) > 0$, then $\dot{\mathcal{L}}(X(t)) < 0$. Since all queues that come before k_0 are empty (so that for all those queues $\dot{x}^{(k)}(t) = 0$) and since $\dot{x}^{(k_0)}(t) = \chi$, we can write

$$\begin{aligned}\dot{\mathcal{L}}(X(t)) &\le \left(\frac{|\chi|}{2C}\right)^{k_0}\left[\chi + \sum_{i=1}^{N-k_0} C\left(\frac{|\chi|}{2C}\right)^i\right]\\ &= \left(\frac{|\chi|}{2C}\right)^{k_0}\left[\chi + \frac{|\chi|}{2}\sum_{i=0}^{N-k_0-1}\left(\frac{|\chi|}{2C}\right)^i\right]\\ &< 0\end{aligned}$$

because the sum is smaller than 2. Thus, $X(t) = 0$ is the only solution[11] of the fluid model for the initial condition $X(0) = 0$. The fluid model is weakly stable and therefore the stochastic system is rate-stable by Theorem 2.

The proof can be extended to the case in which more flows that visit the same station fall in the same priority class. The arguments on which the proof is based are similar to the previous ones, thus we only sketch here the main steps of the proof. If different flows with the same priority visit the same station, we must partition traffic flows in classes, denoted

[11]Assume $\mathcal{L}(t) \ge 0$, $\mathcal{L}(0) = 0$, and $\dot{\mathcal{L}}(t) \le 0$. Consider $\mathcal{L}^2(t) = 2\int_0^t \mathcal{L}(x)d\mathcal{L}(x)$. By definition, $\mathcal{L}^2(t) \ge 0$, but $\int_0^t \mathcal{L}(x)d\mathcal{L}(x) \le 0$. Hence $\mathcal{L}(t) = 0, \forall t$ (see [14]).

by $K_i, i = 1, \ldots, L$, where L is the cardinality of the partition and is characterized by the same ordinal number. We order these classes in such a way that, if $i < j$, then $o_{f_m} < o_{f_n}$ $\forall f_m \in K_i, f_n \in K_j$. Let K_α represent the highest priority group of flows to which corresponds at least a nonempty virtual queue and use now k to indicate any virtual queue belonging to a flow in K_α. Finally, let $i < \alpha$, so that K_i denotes any priority class higher than K_α, and let h indicate any virtual queue belonging to a flow in K_i.

The presence of lower priority virtual queues does not perturb the behavior of $\dot{x}^{(k)}(t)$, because of the SP discipline. Moreover, $\dot{x}^{(h)}(t) = 0$ at every regular point t. Thus, $\dot{t}^{(h)}(t) = \omega^{(h)}$. This allows us to locally study the behavior of $\dot{x}^{(k)}(t)$: indeed, virtual queues belonging to flows in lower priority classes can be ignored, while every $v^{(h)}$ subtracts at its correspondent server $j = L(h)$ an amount of work equal to $\omega^{(h)}$. In other words, the study of local behaviors of $v^{(k)}$ can be reduced to the study of the evolution of queue lengths in an underloaded single priority system in which virtual queues are served according to FIFO. By applying the same Lyapunov function $\mathcal{L}_{FIFO}$ used in [22][12] and denoting with $\mathbb{1}_{K_\alpha}$ a $N \times N$ diagonal matrix that has unit diagonal elements in correspondence of virtual queues belonging to K_α and null elements elsewhere, it is possible to conclude that $\dot{\mathcal{L}}_{FIFO}(\mathbb{1}_{K_\alpha} X(t)) < \chi < 0$. For the lower priority classes K_j, there exists a constant $C > 0$ such that $\dot{\mathcal{L}}_{FIFO}(\mathbb{1}_{K_j} X(t)) < C$, since all the growth rates of queues are finite. Thus, defining $\mathcal{L}(X(t)) = \sum_{i=1}^{L} (|\chi|/2C)^i \mathcal{L}_{FIFO}(\mathbb{1}_{K_i} X(t))$ it results $\dot{\mathcal{L}}(X(t)) < 0$. Then the queueing network is rate-stable, again by Theorem 2. ■

A strict parallelism exists among the stability result for acyclic SP/FIFO queueing networks, and the stability result for networks of GPS schedulers under the CRST constraint proved in [2]. The concept of acyclicity in SP networks, indeed, can be considered as an extension to the case of SP/FIFO disciplines of the concept of CRST defined for GPS schedulers. The extension is, however, non trivial as the following results will show. We discuss next, using AQT arguments, the property of universal stability for acyclic SP/FIFO queueing networks.

Theorem 9: Acyclic SP/FIFO queueing networks are universally stable under any $[\rho, \phi]$ regulated traffic pattern with arbitrary ϕ and average external arrival rate $\rho < 1$, if no pairs of flows traversing the same station have the same priority.

Proof: This proof is similar to the previous one and consists of finding a discrete-time Lyapunov function $\mathcal{L}$ of the queue length vector X_n, such that

$$\mathcal{L}(X_n) \geq 0 \tag{16}$$
$$\mathcal{L}(0) = 0 \tag{17}$$
$$\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n) < 0 \qquad \forall X_n \neq 0. \tag{18}$$

We sample the system evolution every ϕ time units, i.e., we divide time into frames of size ϕ, so that X_n represents the queue state after frame n. In order to simplify the proof, we further suppose that the adversary injects packets at the beginning of each frame. This assumption, however, has no substantial impact on the network stability.

We order virtual queues in the network according to $\mathcal{GO}$. Given any nonnull vector state X_n, indicating with k_0 the first nonnull component of X_n, and letting $L(k_0) = j_0$, we obtain

$$x_{n+1}^{(k_0)} \leq \left\lfloor x_n^{(k_0)} + \rho\phi - \phi \right\rfloor \tag{19}$$

where $\lfloor x \rfloor = \max(x, 0)$. Indeed, no packets in the system at the end of frame n can feed queue k_0 or block packets stored in queue k_0 from accessing the server j_0. Thus, only packets generated in frame $n+1$ may feed queue k_0 or block packets in queue k_0 from accessing the server j_0. Since at most $\rho\phi$ packets requiring service at server j_0 are injected in the network in frame $n+1$, (19) follows. Note that $x_{n+1}^{(k_0)} < x_n^{(k_0)}$ whenever $x_n^{(k_0)} > 0$ (and $\rho < 1$).

All of the queues that come before k_0 are null by construction at the end of frame n and remain null at the end of frame $n+1$. Queues that follow k_0 may increase by at most ϕ. Thus, defining $\mathcal{L}(X_n) = \sum_{k=1}^{N} x_n^{(k)} (1/2\phi)^k$, (16)–(18) are satisfied. In addition, if $X_n = 0$, also $X_{n+1} = 0$, as can be easily seen.

Therefore, for every initial condition X_0 and any $[\rho, \phi]$ regulated adversary, there exists an $M > 0$ such that:[13] $\limsup_{n\to+\infty} \mathcal{L}(X_n) < M$ and thus $\limsup_{n\to+\infty} \|X_n\|_L < \infty$. ■

The extension of the previous result to the case in which several flows share the same priority is impossible, since it was proved [19] that the FIFO service discipline is not universally stable for any $\rho > 0$. Note, indeed, that an acyclic SP/FIFO network reduces to a FIFO network when the same priority is assigned to every flow. The unstable example studied in [18], however, requires that some packets follow closed paths (i.e. trajectories that start and end at the same node). Thus, the reader may wonder whether acyclic SP/FIFO networks can be still proved to be not universally stable under the further restriction that packet routes are selected according to "reasonable" routings.

The following generalization of the example reported in [18] shows that instability still arises in acyclic SP/FIFO networks in which packets are routed according to a shortest path routing. It also shows that the adoption of an acyclic SP/FIFO discipline entails a further reduction of the stability region with respect to a pure FIFO discipline.

Theorem 10: Acyclic SP/FIFO queueing networks are **not** all universally stable for average external arrival rates ρ greater than 0.8, when assuming that all servers have unitary capacity.

Proof: Consider the queueing network in Fig. 6, where queues are represented by numbered circles, and labeled edges represent possible customer routes. In Table I, we specify customer flow routes and priorities (where smaller numbers mean higher priority). We will show that a (ρ, ϕ) adversary

[12] $\mathcal{L}_{\text{FIFO}} = \sum_{j=1}^{J} \sum_{k \in L^{-1}(j)} \int_t^{t+W_j(t)} \pi^{(k)} h(\dot{d}^{(k)}(s)/\pi^{(k)}) ds$ with $h(x) = x \log(x)$ and $W_j(t) = \sum_{k \in L^{-1}(j)} s^{(k)} x^{(k)}(t)$. We recall here that the traffic pattern must satisfy the strong law of large numbers, as stated in Theorem 8. This assumption is required to allow our approach.

[13] By contradiction, suppose $\limsup_{n\to+\infty} \mathcal{L}(X_n) = +\infty$. Then, for every $M_1 > 0$ there exists an n_1 such that $\mathcal{L}(X_{n_1}) > M_1$. Repeating the same reasoning, there exists an $M_2 > \mathcal{L}(X_{n_1})$ and an $n_2 > n_1$ such that $\mathcal{L}(X_{n_2}) > M_2$. Thus, $\mathcal{L}(X_n)$ must have increased between n_1 and n_2. However, this is impossible.

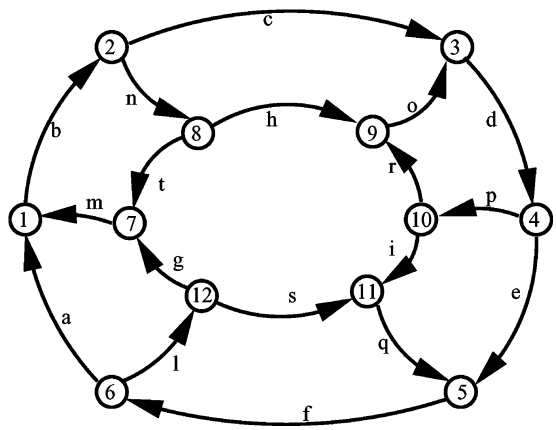

Fig. 6. Acyclic SP/FIFO queueing network that is not universally stable.

exists under which some queue lengths in this network grow indefinitely.

We divide the network evolution into periods. Each period is itself subdivided into phases. Table I also reports the phase in which each flow enters the network. In the following, we consider each phase lasting W_0 time units.

Let us start by examining the network behavior during the first period. We suppose that each server in the network serves flows according to an acyclic SP/FIFO discipline. We suppose that, at the beginning of the period, W_0 packets are stored in the network at node 1 directed to node 9 over path (b, n, h). During the first phase, ρW_0 packets arrive at node 1, directed to node 11, over path (b, c, d, p, i); in addition, ρW_0 packets are injected at node 8, directed to node 9 over path (h). During this phase, all of the W_0 packets $1 \to 9$ move from node 1 to node 8, since they are older than packets $1 \to 11$; thus, at the end of the phase, all packets $1 \to 11$ are still enqueued at node 1. In addition, at the end of the phase, ρW_0 packets belonging to flow $1 \to 9$ are still enqueued at node 8, where the ρW_0 packets belonging to flow $8 \to 9$ could access link h thanks to their higher priority, and only $(1 - \rho)W_0$ packets $1 \to 9$ could be serviced. During the second phase, ρW_0 packets are injected at node 8 directed to node 11 over path (h, o, d, p, i), and ρW_0 packets are injected at node 2 directed to node 3 over path (c). During this phase, all of the higher priority packets belonging to flow $2 \to 3$ reach their destination, along with the packets belonging to flow $1 \to 9$, that can access link h being older than packets $8 \to 11$. While all of the ρW_0 packets $1 \to 11$ have been served at node 1, the residual server capacity at nodes 2 and 8 is exploited by $(1 - \rho)W_0$ packets belonging, respectively, to flows $1 \to 11$ and $8 \to 11$. As a result, at the end of this phase $\rho W_0 - (1-\rho)W_0 = (2\rho - 1)W_0$ packets belonging to flow $1 \to 11$ are enqueued at node 2 and $(2\rho - 1)W_0$ packets belonging to flow $8 \to 11$ are enqueued at node 8. During the third phase, ρW_0 packets are injected at node 3, directed to node 11 over path (d, p, i). Packets belonging to flows $1 \to 11$ and $8 \to 11$ are served at nodes along their paths until they reach node 3. Thus, all packets in the system converge at node 3: here, priority is given to flow $3 \to 11$ so that its ρW_0 packets can reach their destination. The residual server capacity at node 3 is exploited by some packets belonging to flows $1 \to 11$ and/or $8 \to 11$ (they have the same priority, so the packet that is served depends on the FIFO order). At the end of the phase, $W_1 = 2(2\rho - 1)W_0 - (1 - \rho)W_0 = (5\rho - 3)W_0$ packets directed to node 11 are still enqueued at node 3. If $\rho > 0.8$, W_1, the number of packets enqueued at node 3 at the end of the first period, exceeds W_0. Note that the initial condition for the second period closely resembles the initial condition for the first period (the topology clearly exhibits a symmetry under which paths $1 \to 9$ and $3 \to 11$ are equivalent). Thus, there exists an external arrival pattern such that at the end of the second period $W_2 > W_1 > W_0$ packets are stored at node 5 and directed to node 7. Finally, with similar considerations, it is possible to show that there exists an external arrival pattern such that, at the end of the third period, W_3 packets (with $W_3 > W_2 > W_1 > W_0$) are stored at node 1 and directed to node 9 through path (b, n, h). By induction, we can prove that the number of packets left in the network at the end of each period grows indefinitely, thus the network is not universally stable if $\rho > 0.8$. ∎

TABLE I
PRIORITY ASSIGNMENTS FOR FLOWS IN THE NETWORK

Source	Destination	Route	Priority	Phase
1	9	(b, n, h)	1	0
1	11	(b, c, d, p, i)	1	1
8	9	(h)	0	1
2	3	(c)	0	2
8	11	(h, o, d, p, i)	1	2
3	11	(d, p, i)	0	3

V. CONCLUSION

In this paper, we have discussed possible underload instabilities due to GPS and SP schedulers in packet-switching networks, considering scenarios with acyclic packet routes, and service times that vary only according to channel capacities.

Our analysis extends recent results, showing that instability can happen in underloaded queueing networks, loosening the system assumptions in a way that our findings can be applied to the approaches being currently considered for QoS provisioning in the Internet.

Our main results are that: 1) GPS schedulers may be unstable when some of the actual packet rates exceed the GPS rates; 2) the instability of GPS schedulers is not mitigated by AIMD traffic control; 3) SP schedulers are stable, provided that the priority ordering of packet flows does not change from a router to another; and 4) the combination of priority and FIFO schedulers may lead to a weak form of instability for loads larger than 0.8.

The results about the possible instability of GPS schedulers are quite relevant, in light of their possible adoption for the provision of QoS guarantees in the Internet, and indicate that the correct estimation of actual rates of packet flows is crucial. The conditions for the stability of SP schedulers also appear to be quite useful, indicating that the consistent assignment of priorities throughout network domains (such as Internet Autonomous Systems) is essential. Finally, the fact that the combination of priority and FIFO may lead to instability (even if this form of instability is rather weak) for loads above 0.8 shows that the instability phenomena that we have discussed may frequently impact the behavior of packet-switching networks.

REFERENCES

[1] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks—the single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, June 1993.

[2] ——, "A generalized processor sharing approach to flow control in integrated services networks—the multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137–150, Apr. 1994.

[3] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, pp. 1374–1396, Oct. 1995.

[4] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, Assured Forwarding PHB Group, RFC 2597, June 1999.

[5] V. Jacobson, K. Nichols, and K. Poduri, An Expedited Forwarding PHB, RFC 2598, June 1999.

[6] L. Kleinrock, *Queueing Systems*. New York: Wiley, 1976, vol. 2.

[7] J. R. Jackson, "Jobshop-like queueing systems," *Management Science*, vol. 10, no. 1, pp. 131–142, Oct. 1963.

[8] F. Baskett, K. M. Chandy, R. R. Muntz, and F. Palacios, "Open, closed and mixed networks with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, Apr. 1975.

[9] F. P. Kelly, *Reversibility and Stochastic Networks*. New York: Wiley, 1979.

[10] S. H. Lu and P. R. Kumar, "Distributed scheduling based on due dates and buffer priorities," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 1406–1416, Dec. 1991.

[11] J. G. Dai and G. Weiss, "Stability and instability of fluid models for re-entrant lines," *Math. Oper. Res.*, vol. 21, pp. 115–135, 1996.

[12] A. N. Rybko and A. L. Stolyar, "Ergodicity of stochastic processes describing the operation of open queueing networks," *Problems Inform. Transm.*, vol. 28, pp. 199–220, 1992.

[13] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking*, vol. 7, pp. 188–201, Apr. 1999.

[14] J. G. Dai, *Stability of Fluid and Stochastic Processing Networks*, Denmark: Centre for Mathematical Physics and Stochastic, Jan. 1999.

[15] M. Andrews and L. Zhang, "Achieving stability in networks of input-queued switches," in *Proc. IEEE INFOCOM*, vol. 3, Anchorage, AK, Apr. 22–26, 2001, pp. 1673–1679.

[16] M. Ajmone Marsan, E. Leonardi, M. Mellia, and F. Neri, "On the throughput achievable by isolated and interconnected input-queueing switches under multiclass traffic," in *Proc. IEEE INFOCOM*, vol. 3, New York, June 23–27, 2002, pp. 1605–1614.

[17] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, pp. 1260–1272, Aug. 1999.

[18] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu, "Universal stability results for greedy contention-resolution protocols," *J. ACM*, vol. 48, no. 1, pp. 39–69, Jan. 2001.

[19] R. Bhattacharjee and A. Goel, "Instability of FIFO at arbitrarily low rates in the adversarial queueing model," in *Proc. IEEE Foundations of Computer Science (FOCS) 2003*, Cambridge, MA, Oct. 11–14, 2003, pp. 160–167.

[20] D. Gamarnik, "Using fluid models to prove stability of adversarial queueing networks," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 741–746, Apr. 2000.

[21] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 251–260, Feb. 1995.

[22] M. Bramson, "Convergence to equilibrium for fluid models of FIFO queueing networks," *Queueing Syst.*, vol. 22, pp. 5–45, 1996.

[23] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson, "Adversarial queueing theory," *J. ACM*, vol. 48, no. 1, pp. 13–38, Jan. 2001.

[24] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114–131, Jan. 1991.

[25] ——, "A calculus for network delay, part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, pp. 132–141, Jan. 1991.

[26] M. Andrews and L. Zhang, "The effects of temporary sessions on network performance," in *Proc. 11th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA) 2000*, San Francisco, CA, Jan. 9–11, 2000, pp. 448–457.

[27] M. Ajmone Marsan, M. Franceschinis, P. Giaccone, E. Leonardi, F. Neri, and A. Tarello, "Instability phenomena in underloaded packet networks with elastic traffic," in *Proc. IEEE GLOBECOM*, San Francisco, CA, Dec. 1–5, 2003, pp. 2913–2917.

Marco Ajmone Marsan (F'99) holds degrees in electronic engineering from the Politecnico di Torino, Torino, Italy, and the University of California at Los Angeles (UCLA). In 2002, he was awarded an *Honoris Causa* degree in telecommunication networks from the Budapest University of Technology and Economics, Budapest, Hungary.

He is a Full Professor in the Electronics Department, Politecnico di Torino, Italy, and the Director of the Institute for Electronics, Information Engineering and Telecommunications of the National Research Council. He has coauthored over 300 journal and conference papers in the areas of communications and computer science, as well as the two books, *Performance Models of Multiprocessor Systems* (Cambridge, MA: MIT Press) and *Modeling with Generalized Stochastic Petri Nets* (New York: Wiley). His current research interests are in the fields of performance evaluation of communication networks and their protocols.

Dr. Ajmone Marsan received the Best Paper Award at the Third International Conference on Distributed Computing Systems in Miami, FL, in 1982. He is a corresponding member of the Academy of Sciences of Torino. He participates in a number of editorial boards of international journals, including the IEEE/ACM TRANSACTIONS ON NETWORKING.

Mirko Franceschinis (S'00) received the Dr.Ing. and Ph.D. degrees in telecommunications engineering from Politecnico di Torino, Turin, Italy, in 2000 and 2004, respectively.

He currently holds a Postdoctoral position with the Eletronics Department, Politecnico di Torino. From October 2002 to July 2003, he visited the Laboratory for Computer Communications and Applications, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. His main area of interest is performance evaluation of communication networks.

Emilio Leonardi (M'99) was born in Cosenza, Italy, in 1967. He received the Dr.Ing. degree in electronics engineering and the Ph.D. degree in telecommunications engineering both from Politecnico di Torino, Turin, Italy, in 1991 and 1995, respectively.

He is currently an Assistant Professor with the Electronics Department, Politecnico di Torino. In 1995, he visited the Computer Science Department of the University of California, Los Angeles (UCLA); in the summer of 1999 he joined the High Speed Networks Research Group, Bell Laboratories/Lucent Technologies, Holmdel, NJ; in the summer of 2001, the Electrical Department, Stanford University, Stanford, CA; and, finally, in the summer of 2003, the IP Group at Sprint, Advanced Technogies Laboratories, Burlingame, CA. He has coauthored over 100 papers published in international journals and presented in leading international conferences, all of them in the area of telecommunication networks. His research interests are in the field of: performance evaluation of communication networks, all-optical networks, queueing theory, and packet switching.

Dr. Leonardi was the recipient of the IEEE TCGN Best Paper Award for a paper presented at the IEEE Globecom 2002 "High Speed Networks Symposium." He was a member of the program committee of several conferences, including IEEE Infocom, IEEE Globecom, and the IEEE International Communications Conference. He was the Guest Editor of two special issues of the IEEE JOURNAL OF SELECTED AREAS OF COMMUNICATIONS that focused on high-speed switches and routers.

Fabio Neri (M'99) received the Dr. Ing. and Ph.D. degrees in electrical engineering from Politecnico di Torino, Turin, Italy, in 1981 and 1987, respectively.

He is currently a Full Professor with the Electronics Department, Politecnico di Torino. His teaching responsibilities include graduate-level courses on computer communication networks and on the performance evaluation of telecommunication systems. His research interests are in the fields of performance evaluation of communication networks, high-speed and all-optical networks, packet switching architectures, discrete event simulation, and queuing theory. He has coauthored over 100 papers published in international journals and presented in leading international conferences. He leads a research group on optical networks at Politecnico di Torino.

Prof. Neri was general co-chair of the 2001 IEEE Local and Metropolitan Area Networks (IEEE LANMAN) Workshop and general chair of the 2002 IFIP Working Conference on Optical Network Design and Modeling (ONDM).

Alessandro Tarello (S'03) received the Dr.Ing. degree in telecommunications engineering from Politecnico di Torino, Turin, Italy, in 2002, where he is currently working toward the Ph.D. degree.

His research interests are in the field of models for performance evaluation of communication networks and their protocols.